

# Kotlin - Basic Syntax

## Kotlin Program Entry Point

An entry point of a Kotlin application is the **main()** function. A function can be defined as a block of code designed to perform a particular task.

Let's start with a basic Kotlin program to print "Hello, World!" on the standard output:

```
fun main() {  
    var string: String = "Hello, World!"  
    println("$string")  
}
```

When you run the above Kotlin program, it will generate the following output:

```
Hello, World!
```

## Entry Point with Parameters

Another form of **main()** function accepts a variable number of String arguments as follows:

```
fun main(args: Array<String>){  
    println("Hello, world!")  
}
```

When you run the above Kotlin program, it will generate the following output:

```
Hello, World!
```

If you have observed, its clear that both the programs generate same output, so it is very much optional to pass a parameter in **main()** function starting from Kotlin version 1.3.

## print() vs println()

The **print()** is a function in Kotlin which prints its argument to the standard output, similar way the **println()** is another function which prints its argument on the standard output but it also adds a line break in the output.

Let's try the following program to understand the difference between these two important functions:

```
fun main(args: Array<String>){  
    println("Hello,")  
    println(" world!")  
}
```

```

    print("Hello,")
    print(" world!")
}

```

When you run the above Kotlin program, it will generate the following output:

```

Hello,
world!
Hello, world!

```

Both the functions (print() and println()) can be used to print numbers as well as strings and at the same time to perform any mathematical calculations as below:

```

fun main(args: Array<String>){
    println( 200 )
    println( "200" )
    println( 2 + 2 )

    print(4*3)
}

```

When you run the above Kotlin program, it will generate the following output:

```

200
200
4
12

```

## Semicolon (;) in Kotlin

Kotlin code statements do not require a semicolon (;) to end the statement like many other programming languages, such as Java, C++, C#, etc. do need it.

Though you can compile and run a Kotlin program with and without semicolon successfully as follows:

```

fun main() {
    println("I'm without semi-colon")

    println("I'm with semi-colon");
}

```

When you run the above Kotlin program, it will generate the following output:

```

I'm without semi-colon
I'm with semi-colon

```

So as a good programming practice, it is not recommended to add a semicolon in the end of a Kotlin statement.

## Packages in Kotlin

Kotlin code is usually defined in packages though package specification is optional. If you don't specify a package in a source file, its content goes to the default package.

If we specify a package in Kotlin program then it is specified at the top of the file as follows:

```
package org.tutorialspoint.com
```

```
fun main() {  
    println("Hello, World!")  
}
```

When you run the above Kotlin program, it will generate the following output:

```
Hello, World!
```

## Quiz Time (Interview & Exams Preparation)

**Q 1 - Kotlin main() function should have a mandatory parameter to compile the code successfully:**

- A - True
- B - False

**Q 2 - What will be the output of the following Kotlin program**

```
fun main() {  
    println("1"); println("2")  
}
```

- A - This will give a syntax error
- B - It will print 12
- C - 1 followed by 2 in the next line
- D - None of the above

**Q 3 - Which of the following statement is correct in Kotlin**

A - A Kotlin program must have a **main()** function

B - A Kotlin program can be compiled without a **main()** function

C - It is mandatory to have a **print()** or **println()** functions in a Kotlin program

D - All statements are correct from Kotlin programming point of view

mohamedsohel.co.in